Machine Learning Engineer Nanodegree

Capstone proposal

Grasp quality prediction of a robotic hand

Uljan Sinani – 09/07/2020

Domain background

Often, we take for granted our human capability to grasp objects and smoothly manipulate them. As trivial as it may seem for us, it is not the case for a robot. Industrial environments are a common example where we desire the robots to perform tasks close to human capability. Although, robotic manipulators may have superior strength, and might not be endangered from harsh environments, they lack the ability to accurately grasp and handle object properly. Therefore, to address these issues that robotics field is facing, a machine learning approach could come to the rescue to providing proper domain-related solution.

Problem statement

In industrial environments time and energy are the most important variables to optimize for. The reasons they are important is related to the amount of time it takes for a robot to perform a certain task and as a result a certain amount of energy will get spend performing that action. The latter will get worse if the task must be repeated many times and even worse when it is not accomplishing it properly. Thus, in industry where potentially hundreds or more robots are installed to perform certain tasks the success rate to accomplish a quality grasp could make or break a task. On top of that, if a certain output per time is required to be met, then the failure rate to grasping objects should be smallest possible. Based on some research in industry the following reasons are shown to be related with the importance of robot grasping.

- Saving energy by performing the action only once
- Grasping effectively the object without causing any damage
- Saving time by quickly grasping the object
- Flexibility, adapting the grasp force according to the nature of the object (i.e. soft, hard, fluffy etc.)

Datasets and inputs

The test dataset that will be acquired for this problem is based in a real commercial company that has released publicly the data from a grasping robotic hand. Shadow robotics' dataset obtained from Kaggle will serve as input data. It contains of a single CSV file that contains the number of test experiments performed with the aim to grasp the object using robot's End-effector. In addition, in the table below is shown a quick peek of the data regarding the number of experiments it contains, robustness and other variables.

[16]:
```python
#ignoring the first row (header)
# and the first column (unique experiment id, which I'm not using here)
dataset = pd.read_csv("/kaggle/input/shadow_robot_dataset.csv")
dataset.head(10)
```

Out[16]:

	experiment_number	robustness	H1_F1J2_pos	H1_F1J2_vel	H1_F1J2_eff	H1_F1J3_pos	H1_F1J3_vel	H1_F1J3_eff	H1_F1J1_pos	H1_F1J1_vel	...	H1_F2J1_pos	H1_F2J1_vel
0	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.118209	6.838743	1.454113	0.302276	-18.738705	0.0	-0.032352	0.127232	...	0.109246	0.042166
1	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.152945	5.997176	1.098305	0.308893	-14.173090	0.0	-0.027381	0.273711	...	0.105656	-0.130178
2	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.162168	5.302321	0.999142	0.314331	-13.093510	0.0	-0.025808	0.184343	...	0.103620	-0.162815
3	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.137684	6.504519	1.256002	0.304333	-16.948796	0.0	-0.027398	0.121100	...	0.106332	-0.186364
4	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.161747	4.899113	0.999313	0.315815	-13.700695	0.0	-0.025698	0.079876	...	0.104104	-0.216307
5	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.142037	6.244418	1.209869	0.306419	-16.266108	0.0	-0.027343	0.144840	...	0.105687	-0.166225
6	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.166825	5.202907	0.951563	0.313909	-12.279431	0.0	-0.025928	0.199046	...	0.103709	-0.183254
7	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.163068	3.637815	0.973553	0.319590	-10.473125	0.0	-0.025502	0.183853	...	0.103900	0.250393
8	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.161140	5.029976	1.006698	0.315507	-13.201170	0.0	-0.026505	0.137231	...	0.103814	0.254959
9	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.162738	4.633381	0.986761	0.309445	-13.416015	0.0	-0.025182	0.043798	...	0.105056	-0.216996

10 rows × 30 columns

Figure 1 Dataset content - first ten rows

The dataset contains 29 Columns and 992640 rows with data for each finger join value of the end-effector. The columns' descriptions appearing in the data set have abbreviated terms related to hand, fingers and joints. For instance, H1 stands for Hand one, F1 for finger one (three fingers per hand), J1 for joint one (three joints per finger), etc.

[20]:
```python
dataset.loc[0]
```

Out[20]
```
experiment_number
 robustness
 H1_F1J2_pos
 H1_F1J2_vel
 H1_F1J2_eff
 H1_F1J3_pos
 H1_F1J3_vel
 H1_F1J3_eff
 H1_F1J1_pos
 H1_F1J1_vel
 H1_F1J1_eff
 H1_F3J1_pos
 H1_F3J1_vel
 H1_F3J1_eff
 H1_F3J2_pos
 H1_F3J2_vel
 H1_F3J2_eff
 H1_F3J3_pos
 H1_F3J3_vel
 H1_F3J3_eff
 H1_F2J1_pos
 H1_F2J1_vel
 H1_F2J1_eff
 H1_F2J3_pos
 H1_F2J3_vel
 H1_F2J3_eff
 H1_F2J2_pos
 H1_F2J2_vel
 H1_F2J2_eff
 measurement_number
Name: 0, dtype: object
```

```python
dataset.describe()
```

	robustness	H1_F1J2_pos
count	992641.000000	992641.000000
mean	77.223840	0.137189
std	53.068893	0.169908
min	0.000001	-5.235986
25%	15.384972	0.070505
50%	94.795925	0.112284
75%	118.369791	0.220674
max	2933.584361	4.886928

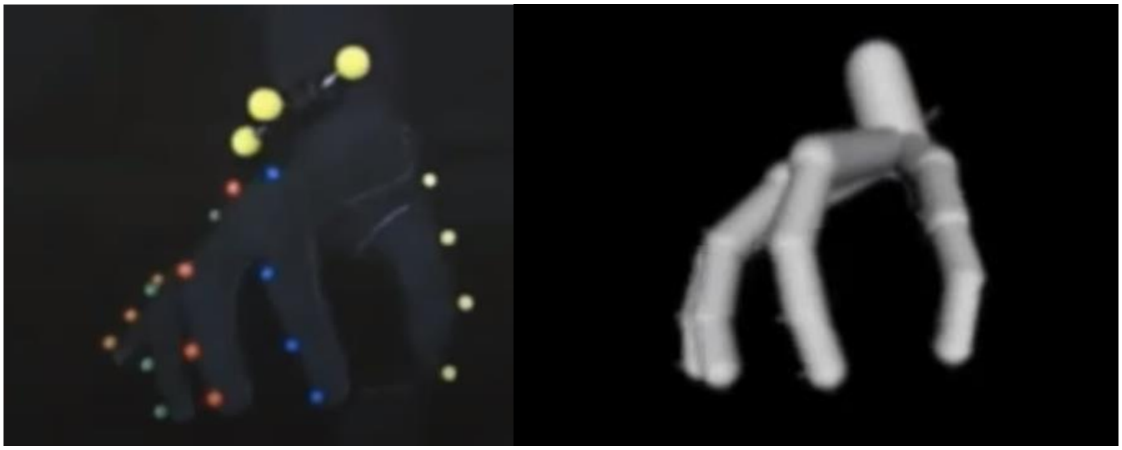
8 rows × 29 columns

Figure 2 Data sample of CSV file

Given that the dataset is obtained using Smart Grasping Sandbox. The values to calculating the grasp robustness are determined by the variation of the distance between the pal and the ball while the robot is performing an action such as shaking the object.

Solution statement

To solving the goal achieving a quality grasp it is first important to define what is a good grasp. Kinematically it means the joints of the end effector will have close to zero angular velocity, and zero acceleration compared to neighboring joints relative to the object. The more joints are engaged during the grasp of an object the better the grasp.



Having defined the notion of a good grasp and possessed the data the next stage I plan to list the most relevant variables – generated from the simulated sandbox – and then use those for the neural network.

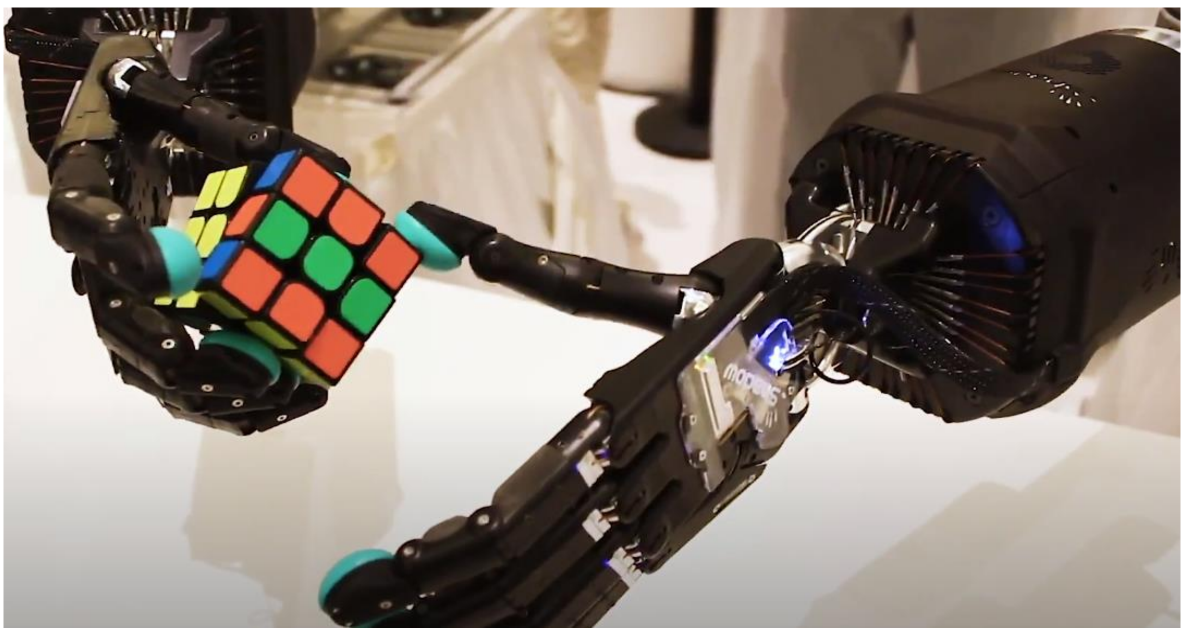


Table 1 Example of good grip

An example of values I plan to take into consideration for the neural network is shown in the table below.

Variable description	Peak	Mean
Angular velocity	rad/s	rad/s
Linear velocity	m/s	m/s
Angular acceleration	Rad/s^2	Rad/s^2
Linear acceleration	m/s^2	m/s^2

Table 2 End-effector example of magnitude values to be determined

The first network I will try to apply in order estimate the grasp quality will be LSTM neural network. The reason related to the fact that this type of network could accept sampled inputs quantized in time. As such, the dataset provided in Kaggle shows to have the datapoints, time steps and other features which are the variable such as the velocity of finger joints etc.

Benchmark model

For the model benchmark I plan to apply initially LSTM and determine the accuracy of the grasp. Then I will compare it with other Machine Learning frameworks such as Keras, SKlearn in order to find which one performs best for a quality grasp.

Evaluation Metrics

The main metrics to evaluate the model will the predicted grasp robustness in respect to the number of times the robotic hand tries to grasp an object. The raw data that will feed the network is expected to be a vector of values from robustness of several points, but in the it will constitute of a single value for the elements/joints of the End-effector.

Project design

To start the project, initially I plan to load and explore the nature of the data. Then, the dataset will be checked for any data inconsistencies and/or missing data, in which case they will be corrected. Next, splitting the dataset into train and test files will create four separate sets X_train, Y_train, X_test and Y_test. The following step will the creation of the Neural Network Model into which the data will be fed. I initially plan to implement LSTM, and then load it into the workspace to prepare for the training step. The final step I plan is to evaluate the model, where the output will be the predicted accuracy of the object grasp.

References

[1] Carlos Rubert, Daniel Kappler, Jeannette Bohg and Antonio Morales: Grasp success prediction with quality metrics

[2] Jialiang (Alan) Zhao, Jacky Liang, and Oliver Kroemer: Towards Precise Robotic Grasping by Probabilistic Post-grasp Displacement Estimation

[3] Alex Keith Goins: thesis work

[4] https://www.kaggle.com/ugocupcic/grasp-quality-prediction/

[5] https://www.youtube.com/watch?v=aTMFBnEdA4I